A Factor-Based Model for Flight Delay Prediction

Team 77 - Gauri Kapse, Subham Biswas, Kartik Tripathi, Rudra Gupta, Saurabh Tahiliani, Alvin Handoko

1 INTRODUCTION

1.1 Overview

Flight delay prediction is a known analytics problem. However, most existing models for delay prediction focus on one of two problems: predicting the extent of a delay for a particular flight, or providing insights into factors affecting the extent of a delay. Additionally, most prediction models work on a real-time or near-real-time basis, i.e., their prediction horizons are limited to 24-48 hours in the future. There is currently a lack of a single tool that can both predict a flight delay and provide insights into the impacting factors, and do this over a slightly longer horizon (Q2).

In this project, we aimed to fill this gap by developing a scalable factor-based model to accurately predict flight delays and highlight the most influential factors contributing to the delay (Q1, Q3). We leverage this model in a tool that displays predicted delays over a slightly longer horizon, as we expect earlier predictions to be more actionable.

Our tool will also provide alternate options. Passengers could use our predictions and alternate recommendations to assist with travel planning, and airlines could use our insights to identify operational areas for optimization (Q4, Q5a).

1.2 Literature Review

A broad body of work has gone into solving this problem. Hsiao et al define a delay as any deviation between actual and scheduled arrival/departure time in excess of 15 minutes (Patgiri et al, 2020).

While some papers have proposed network models to assess delay propagation across airports (Qu and Zhang, 2021; Guvercin et al, 2021), others have limited their scope to specific airports (Zhang et al, 2021) or presented case studies of specific airlines (Wu et al, 2018). In 2016, Truong presented a study, where airlines were segmented based on their on-time arrival rates, and in 2021 Güvercin et al presented a similar clustering for airports in a network. Other papers have demonstrated practical methods for data storage to enhance performance in production environments (Elsayed et al, 2023).

For the prediction task specifically, models proposed to predict the delay amount range from simple factor-based regressions (Hsiao and Hansen, 2006), to random forests (Gui et al, 2020; Li and Jing, 2021), decision trees (Manna et al, 2017), and complex neural network architectures (Wang et al, 2022; Gui et al, 2020). The temporal effects of flight delays have also been accounted for using autoregressive models that account for trends and seasonality (Tu et al, 2008). There are also classification models using statistical learning paradigms like SVM, that simply predict whether there will be a delay (Wu et al, 2019). All of these and other papers have presented creative feature engineering approaches (Wang et al, 2022; Zoutendijk and Mihaela, 2021; Jiang et al, 2020), visualizations (Cheng et al, 2019; Kalliguddi and Leboulluec, 2023), data cleaning techniques and modifications to well-established algorithms (Tu et al, 2008).

Most papers that we've read have had a focus on delay prediction accuracy, which could be sufficient from the point of view of the passenger who is only interested in knowing if their chosen flight will be delayed. However, the prediction window is usually limited to a day or two in the future, limiting the actions a passenger can take when presented with the prediction. We will combine our learnings to build a model that can make predictions over a longer period, and also present passengers with actionable alternatives in the event of a long delay.

Additionally, most papers so far have built models with either passengers or airlines in mind. We attempted to build an interface that can benefit both parties by focusing on machine learning interpretability. Such an interface can provide actionable insights to airlines and/or airports operations management, aiding them with the choice of mitigation measures to minimize delay impacts.

2 METHODS

2.1 Modeling and Interpretability

2.1.1 Dataset Description and Cleaning: To limit the scope of the experiment to the current semester (Q8), we focused exclusively on flight arrivals and departures at Dubai International Airport, using an open dataset provided by Dubai Pulse in combination with a weather dataset provided by the US National Centers for Environmental Information.

The flights dataset was retrieved as two CSV files, one for the arrivals data and one for the departures data. Each file contained information pertaining to each flight that arrived at/departed from DXB. The weather dataset had a per-day resolution and contained data recorded by the measuring station located at DXB. The weather data was joined with flight data on the date of the record. For each feature with missing values, we imputed the missing values with the mode of that column.

The data cleaning and feature engineering techniques employed were largely the same for both arrival and departure datasets. Due to inconsistencies in reporting, we pruned our data to only include flight information for approximately 2.5 years (2020-12-26 to 2023-03-13). For arrivals, we calculated the delay as the difference between the actual and scheduled in-block time (i.e., the time at which an airplane docks at the gate), and for departures we calculated it as the difference between the actual and scheduled off-block times.

We considered flight records with delays that were 3 standard deviations farther from the mean as outliers, and removed them from our data. Based on the calculated delays, we found that our dataset was imbalanced – the highest frequency flight delay observed was around 2 minutes, and higher delays were less represented. Since reporting a 2 minute delay is not useful to an end-user, we bucketed our flights into 3 bins – a flight with a delay under 15 minutes was categorized as on time, a flight with a delay of between 15 to 45 minutes was categorized as having a delay under 1 hour, and all other flights were categorized as having delays over 1 hour. Negative delays (early arrivals/departures) were also assigned to the first bin. This non-isochronous binning helped us balance our dataset and turned our problem from a regression into a classification.

To avoid multicollinearity, we dropped highly correlated or redundant features. For example, between daily minimum, maximum and mean temperatures, only the mean temperature was retained. A few features were reported in both English and Arabic – the Arabic versions were dropped. Similarly, the dataset included certain features reported according to both IATA and ICAO conventions - only one of two were retained. However, we still retained a few correlated weather features like mean temperature and dewpoint, as these translate into different physical problems for airport operations to handle.

Unhelpful features like primary keys, latitude or snow depth were also dropped.

2.1.2 Feature Engineering: Our first challenge was handling the large number of unique values in non-numerical columns. Over 146 and 170 airlines were represented in our arrivals and departures datasets respectively – to get around this, we tallied the proportion of flights in/out of DXB that each airline operated, and only the airlines that contributed to the top 80% of the flights in/out of DXB retained their identities. The smaller airline operations were grouped under the umbrella of 'Other', which reduced the total number of unique airlines to 11 in our arrivals dataset and 10 in our departures dataset. We repeated this same process for unique flight numbers.

Certain flights, if operated jointly by two airlines, also had a joint flight number. Since most flights are not jointly operated, we did not retain the joint flight numbers, but rather converted this column into an indicator variable which took the value of 1 if the flights were jointly operated and 0 otherwise.

Additionally, we extracted the year, month, day of month and day of week from the publicly scheduled date-time for each flight, and used these in our analysis instead of the date directly.

Then we split our data into training and test sets - the training set contained data collected on or prior to October 1st 2022, and the test set contained the remaining data.

2.1.3 Delay Prediction Modeling and Interpretability: For delay prediction, we trained a CatBoost Classifier (developed by the team at Yandex) as it automates the conversion of categorical features into numerical features.

We used the feature importance attribute of the fitted model to identify the top 10 most important features across our data. For the arrivals dataset, the top 10 features accounted for 85.3% of the total variability in the data, and for the departures dataset, they accounted for the top 78.6% of the variability in the data. There was significant overlap between the important features for both arrivals and departures predictions.

At a local level, we employed the SHAP (SHapley Additive exPlanations) algorithm to interpret why our model made a certain prediction for a given data point using Slundberg's library, developed specifically for tree-based ensemble methods.

2.1.4 Recommendations: One of the main functions of our app is providing alternative options to users who look up a specific flight. Initially we attempted to generate these recommendations using counterfactuals. For this, we used the open-source DiCE Python library. However, after some experimentation we found that the generated counterfactual examples were infeasible – flights that don't exist were being predicted.

Hence, we simplified our approach. Now, we just filter flights from our dataset which are predicted to be on time, are going to/coming from the user's chosen location, and are scheduled within a 10 day interval centered on the user's chosen date.

2.1.5 Hardware: All data cleaning, EDA and modeling was performed on shared notebooks in Google Colab, using a runtime with a system RAM of 12.7GB and a Tesla T4 GPU for training the CatBoost Classifier.

The app is configured to run on any commodity computer.

2.2 App Infrastructure

Precomputed delay predictions, feature importances and recommendations are stored in a data-store. These predictions and data are loaded into memory on start.

The home page of the app consists of a form where the user provides their chosen flight number and preferred date of travel (Figure 1). This form includes checks to ensure validity of the inputs. If the user input is invalid, an error message is displayed, stating that the selected flight and date combination is not available in our database. If the user input is valid, the user may click the submit button.

The input is delivered to the backend using REST API calls. Once the API receives inputs for prediction, the corresponding records are fetched from the data-store, and the prediction dashboard is rendered.

The dashboard page shows the selected flight's details, along with the our delay prediction for said flight. It also displays recommendations for similar flights (Figure 2). At the bottom, we also render two plots displaying the past performance of the flight and the factors contributing to our prediction (Figure 3).

The backend server of the web-app is built in Flask and the front-end is built using HTML, CSS and JavaScript. We have used Bootstrap to apply styling and D3.js to build our visualizations.

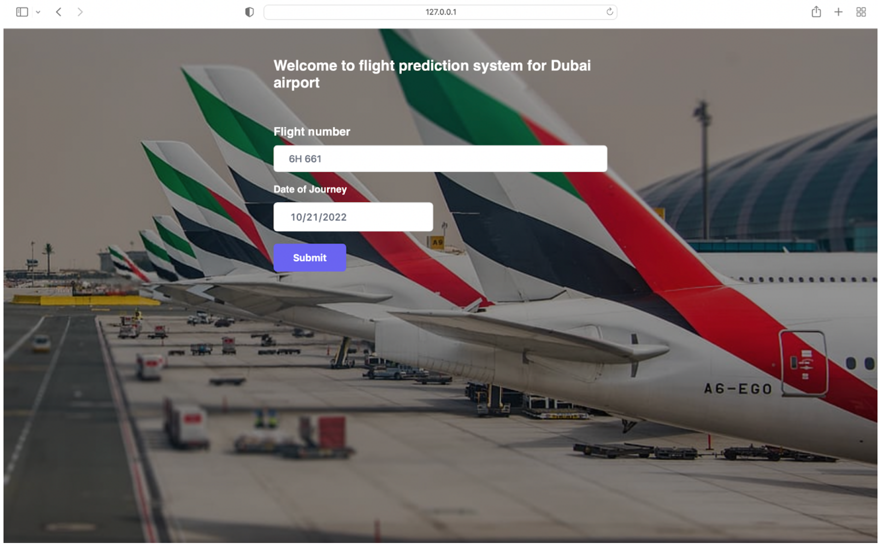
3 RESULTS AND DISCUSSION

3.1 Model Performance

Our arrivals and departures models reported accuracies of 75% and 81% when evaluated on their respective test sets. However, since our datasets were imbalanced (there were more records in the 'No delay' class than both the other classes combined), using accuracy to report model performance is insufficient.

We believe that the F1 score is an appropriate measure, as we wish to give equal weight to precision and recall. Since our tool is expected to be geared towards both airlines and passengers, we want to avoid hurting airline revenues due to erroneous delay predictions, so we want a high precision. Similarly, we want our predictions to capture as many true delays as possible to avoid corroding user trust, so we want a high recall.

We have reported the class-wise precision, recall and F-1 scores in tables 1 and 2 for the arrivals

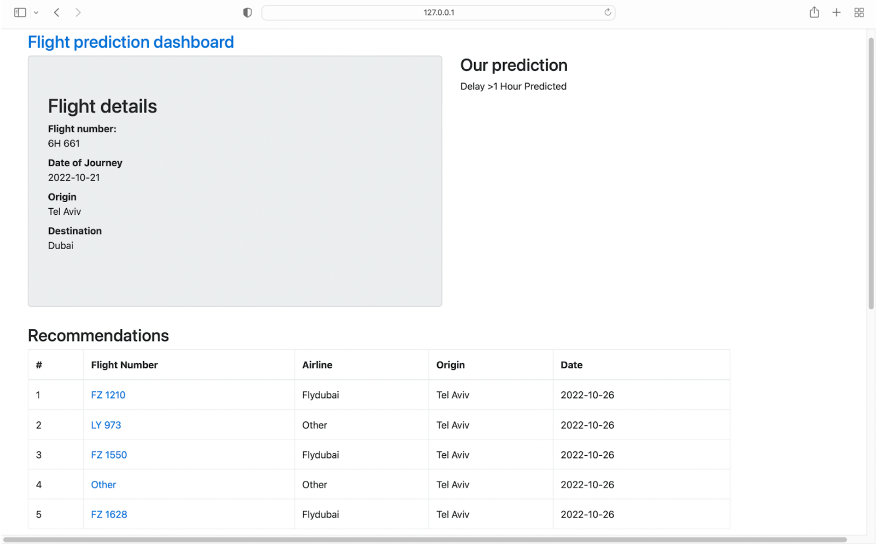


Welcome to flight prediction system for Dubai airport

Flight number
6H 661

Date of Journey
10/21/2022

Figure 1: The user enters their flight information in a form on the landing page of the web app



Flight prediction dashboard

Flight details

Flight number:
6H 661

Date of Journey
2022-10-21

Origin
Tel Aviv

Destination
Dubai

Our prediction

Delay >1 Hour Predicted

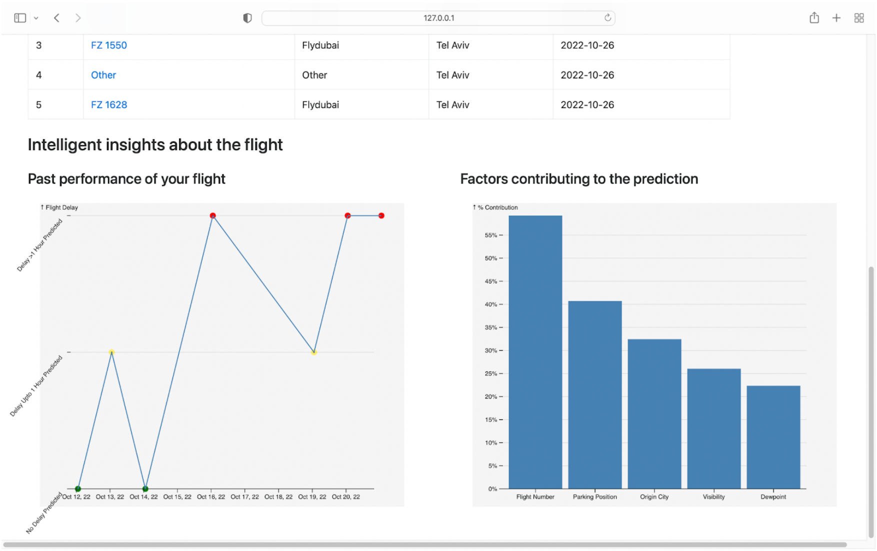
Recommendations

#	Flight Number	Airline	Origin	Date
1	FZ 1210	Flydubai	Tel Aviv	2022-10-26
2	LY 973	Other	Tel Aviv	2022-10-26
3	FZ 1550	Flydubai	Tel Aviv	2022-10-26
4	Other	Other	Tel Aviv	2022-10-26
5	FZ 1628	Flydubai	Tel Aviv	2022-10-26

Figure 2: Results - The app displays flight details, a delay prediction and similar flight recommendations for the user's chosen flight

and departures models respectively. Generally, both models report F1-scores over 85% for the 'No delay' class, but perform poorly for the other two classes. This can be attributed to the imbalanced nature of the data, especially for the 'Delay over 1 Hour' class, where we only have a few hundred records available.

We don't anticipate this being a major usability issue for our tool, because delays over 1 hour are rare. However, in a future iteration of our app, we may include a caveat everytime we predict a delay greater than 1 hour.



3	FZ 1550	Flydubai	Tel Aviv	2022-10-26
4	Other	Other	Tel Aviv	2022-10-26
5	FZ 1628	Flydubai	Tel Aviv	2022-10-26

Intelligent insights about the flight

Past performance of your flight

Factors contributing to the prediction

Figure 3: Results - The app also displays trivia about the flight

Class	Precision	Recall	F-1 Score
No delay (9536)	0.82	0.97	0.89
Delay Under 1 Hour (2051)	0.50	0.14	0.21
Delay Over 1 Hour (249)	0.11	0.00	0.01

Table 1: Arrivals Classifier Performance Metrics - the number of records in each class is indicated in the parentheses

Class	Precision	Recall	F-1 Score
No delay (9892)	0.80	0.90	0.85
Delay Under 1 Hour (2708)	0.39	0.24	0.30
Delay Over 1 Hour (245)	0.04	0.00	0.01

Table 2: Departures Classifier Performance Metrics - the number of records in each class is indicated in the parentheses

3.2 Model Interpretability

Features	Importance (Departures)	Importance (Arrivals)
Flight Number	15.06	16.32
Origin / Destination	12.25	17.75
Parking Position	9.39	9.89
Aircraft Type	8.68	8.63
Airline	7.52	6.58
Mean Temperature	6.78	6.55
Dewpoint	5.14	5.16
Windspeed	5.06	4.20
Day of Month	4.97	5.39
Month	4.45	4.79

Table 3: Global Feature Importances

The top 10 features representing the highest variability in the data and their respective feature importances are displayed in table 3. Both models report the same features in slightly different orders, with the flight number and the origin/destination as the top two factors in delay prediction.

Feature Names	Feature Values	Feature Importances
Airline	Saudi Arabian Airlines	-0.43
Flight Number	SV 551	0.20
Destination City	Jeddah/King Abdulaziz Intl	-0.19
Aircraft Type	333	-0.16
Terminal	1	0.15

Table 4: SHAP - Local Interpretability Example

Table 4 displays an example of a local inerpretability result given by SHAP - this summarizes how each feature contributes to an individual prediction. Positive Shapley values indicate that these features had an increased impact on the prediction and negative Shapley values indicate that these features had a reduced impact on the prediction. However, since our features are not completely independent (for example, temperature and dewpoint are linked), the Shapley values may not be completey accurate.

3.3 User Studies

Since this was just a semester-long project, our app is currently not portable enough for a UX study. We cannot share it with the class without also sharing all our code, so we have only performed preliminary testing on our own computers (Q9). If we were to productionize our app, ideally we would measure customer satisfaction via the adoption rate, post-travel surveys from passengers, and cost-savings reports by airlines (Q5b).

3.4 Current Limitations and Future Improvements

Currently, we are using the CatBoost model out of the box. If we continue working on this project, we may perform some hyperparameter tuning to find a better model. Similarly, we would try to rebuild the recommendation system with interpretML's DiCE library with added constraints to ensure the feasibility of our counterfactual examples.

Our current setup involved conducting all analysis on Google Colab and exporting a CSV with predictions. This CSV was used to configure the backend and the frontend of the app. However, if we were to productionize our app, we would need to find a way to integrate the model directly in the app, so that users may call for predictions in real-time.

For this to happen, we would need to procure more data, the cost of which depends on the source. Dubai Pulse, which is where our current data comes from, also provides API-based access to its data. This costs between 420AED to 12600 AED per month (USD 115 to 3430/month).

Additionally, since the aircraft parking positions are important features for both our models, we may have to work closely with Dubai Airports to obtain this information ahead of time – the availability of this information and the weather forecasts are the two main limiting factors in the expected length of our prediction windows.

Additional costs associated with productionizing our solution will depend on a few factors, including the prediction and interpretability model retraining frequencies (to accommodate the data and model drift), server costs and man hours (Q7). Since our current model is trained on only one year's data, it is exposed to irregularities outside that scope (Q6).

We would also look to improve the design of the user interface.

4 DISTRIBUTION OF EFFORTS

All members have contributed a similar amount of effort.

5 REFERENCES

(1) Cheng, S., Zhang, Y., Hao, S., Liu, R., Luo, X., Luo, Q. (2019). Study of Flight Departure Delay and Causal Factor Using Spatial Analysis. Journal of Advanced Transportation, 111,
doi: 10.1155/2019/3525912.

(2) Elsayed, A., Shaheen, M., Badawy, O. (2023). Caching Techniques for Flight Delays Prediction in Big Data Using SparkR. aast.edu. Retrieved February 14, 2023

(3) Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., Zhao, D. (2020). Flight Delay Prediction Based on Aviation Big Data and Machine Learning. IEEE Transactions on Vehicular Technology, 69(1), 140-150,
doi: 10.1109/TVT.2019.2954094.

(4) Güvercin, M., et al (2021). Forecasting Flight Delays Using Clustered Models Based on Airport Networks, IEEE Transactions on Intelligent Transportation Systems, 22(5), 3179-3189,
doi: 10.1109/TITS.2020.2990960.

(5) Hsiao Hansen. (2006). Econometric Analysis of U.S. airline flight delays with time-of-day effects. Transportation Research Record: Journal of the Transportation Research Board, 1951(1), 104–112,
doi: 10.1177/0361198106195100113.

(6) Jiang, Y., Liu, Y., Liu, D., Song, H. (2020). Applying machine learning to aviation big data for flight delay prediction. IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 665–672.

(7) Kalliguddi & Leboulluec. (2017). Predictive Modelling of Aircraft Flight Delay. Universal Journal of Management, 5(10), 485–491,
doi: 10.13189/ujm.2017.051003.

(8) Li Jing. (2021). Generation and prediction of flight delays in Air Transport. IET Intelligent Transport Systems, 15(6), 740–753,
doi: 10.1049/itr2.12057.

(9) Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., Barman, S. (2017) A statistical approach to predict flight delay using gradient boosted decision tree, 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), 1-5,
doi: 10.1109/ICCIDS.2017.8272656.

(10) Patgiri, R. et al (2020, February 17). Empirical Study on Airline Delay Analysis and Prediction. arXiv.org. doi:
10.48550/arXiv.2002.10254

(11) Qu, J., Zhang, J. (2021). Delay propagation evaluation of flight chain model based on big data visual analytics, IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 167-174.

(12) Truong, D. (2016). Developing Airline Segmentation Based on The On-time Performance. International Journal of Aeronautical Science Aerospace Research, 131-140,
doi: 10.19070/2470-4415-1600016.

(13) Tu, Y., et al (2008). Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern. Journal of the American Statistical Association, 103(481), 112–125.
doi: 10.1198/016214507000000257.

(14) Wang, F., Bi, J., Xie, D., Zhao, X. (2022). Flight delay forecasting and analysis of direct and indirect factors. IET Intelligent Transport Systems, 16(7), 890–907. doi:
10.1049/itr2.12183.

(15) Wu, W.,et al. (2019). An improved SVM model for flight delay prediction, 2019 IEEE/ AIAA 38th Digital Avionics Systems Conference (DASC), 1-6, doi:
10.1109/DASC43569.2019.9081611.

(16) Wu, W., Wu, C., Feng, T., Haoyu, Z., Qiu, S. (2018). Comparative Analysis on Propagation Effects of Flight Delays: A Case Study of China Airlines. Journal of Advanced Transportation, 1-10, doi: 10.1155/2018/5236798.

(17) Zhang, J., Peng, Z., Yang, C., Wang, B. (2021). Data-driven flight time prediction for arrival aircraft within the terminal area. IET Intelligent Transport Systems, 16(2), 263–275.

doi: 10.1049/itr2.12142.

(18) Zoutendijk Mihaela (2021). Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem. Aerospace 8:152. doi: 10.3390/aerospace8060152.

(19) Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67. doi: 10.1038/s42256-019-0138-9